Implementation of the IsGroupOf PDU for Network Bandwidth Reduction

Shirley Pratt and David Pratt
Naval Postgraduate School
Computer Science Department
Monterey, CA
[pratts | pratt]@cs.nps.navy.mil

Lawrence Rieger
US Army Training and Doctrine Command
Deputy Chief of Staff for Simulations and Analysis
Fort Monroe, VA
riegerl@emh10.monroe.army.mil

Leroy Jackson
TRADOC Analysis Center - Monterey
Naval Postgraduate School
Monterey, CA
jacksonl@mtry.trac.nps.navy.mil

KEYWORDS

Aggregation, Bundling, Protocol Data Unit, LAN, WAN

Abstract

This paper describes the implementation of a proposed IsGroupOf protocol data unit in a series of simulation experiments designed to test its effectiveness in: 1) significantly reducing wide area network bandwidth requirements during multi-site distributed simulation exercises, and 2) providing entity level representations of grouped units which is needed by wide area sensor simulations. These experiments involve the use of a network gateway to group units together using IsGroupOf protocol data units which are subsequently sent onto a wide area network. Our results show that use of the IsGroupOf protocol data units can reduce wide area network bandwidth by as much as 80-90%. Individual entities of a group are also still able to be relatively accurately represented on visual simulators, which is sufficient resolution for wide area sensor simulations. While the work presented here deals with the development and implementation of Distributed Interactive Simulation (DIS) protocol data units, the same analysis and distribution mechanisms can be applied to other emerging protocols.

INTRODUCTION

Numerous techniques have been employed to reduce the network, graphics and computational overload associated with conducting large-scale Distributed Interactive Simulation (DIS) exercises. These include the use of various network partitioning schemes, static representation of numerous entities as a single unit, or part-time joining or aggregation of entities. Historically, one of the major problems with using geography based network partitioning schemes has been the representation of wide area sensors. Wide area sensors, by virtue of the large number of network groups they subscribe to, effectively negate the benefits of using such partitioning schemes. Constructive models, which typically use static unit level representations, permit interactions of large scale populations, but details associated with entity level interactions are lost.

Part-time joining or aggregation of multiple entities has been successfully implemented to reduce loads during DIS exercises using IsPartOf PDUs (Patterson and Maxel, 1995) and Aggregate PDUs (Foss and Franceschini, 1995). While entities are joined or aggregated, the representation of individual entities is temporarily lost. For many cases, the entities can be easily represented with respect to some other entity, for example, wingstores on an aircraft, aircraft parked on a carrier, tanks moving in a fixed formation. However, when the joined or aggregated entities behave independently or un-

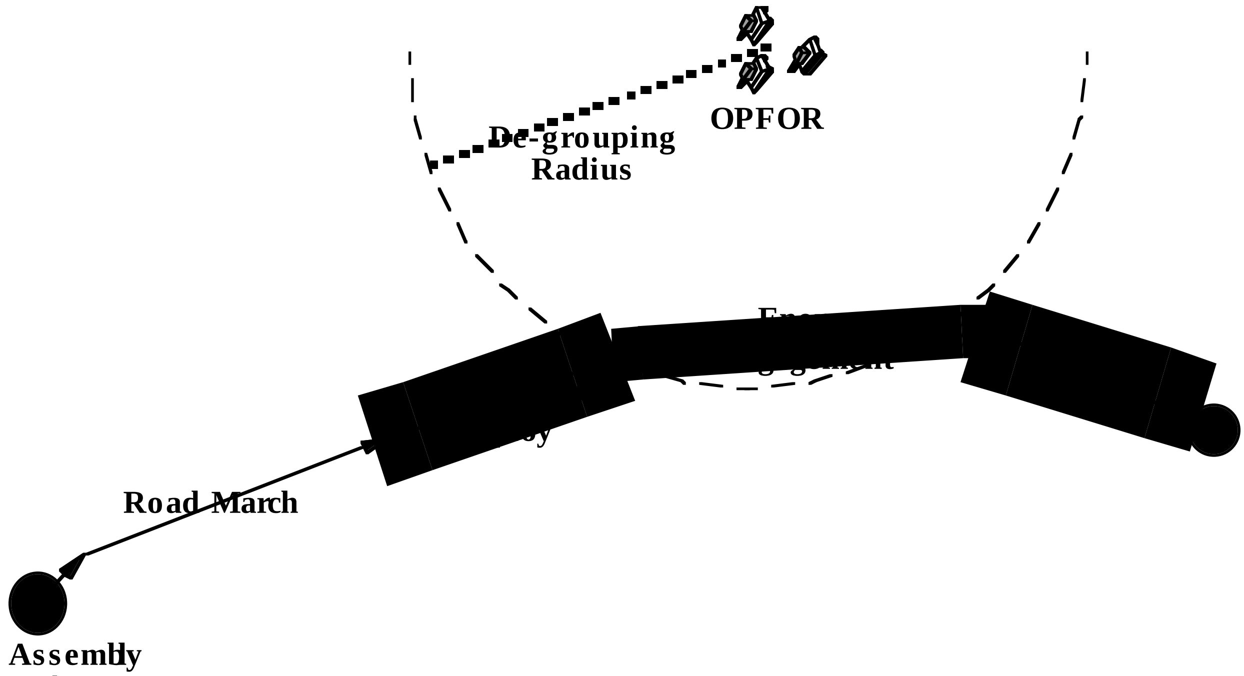

Figure 1. Scenario for IGO PDU Experiments.

predictably, they must usually be split apart again degrading simulation performance.

The rationale for using the IsGroupOf (IGO) Protocol Data Unit (PDU), as presented in our earlier paper (Pratt and Pratt, 1996), is two-fold: 1) to significantly reduce Wide Area Network (WAN) bandwidth requirements during multi-site DIS exercises, and 2) to allow the entity level representation necessary for wide area sensors. The IGO PDU is an evolution of the IsPartOf (IPO) PDU more suited for the representation of grouped units. In this discussion, a group unit still maintains the representation of the individual entities, whereas, a joined or aggregated unit represents only the unit level and individual entity information is lost. By constructing a network group which contains the IGO PDU generators, wide sensors can still effectively represent a correct view of the simulated battlefield. This paper describes the implementation of the IGO PDU in a series of simulation experiments designed to test its effectiveness in allowing entities to behave independently or unpredictably while remaining grouped.

IMPLEMENTATION

The design of the experiments to implement IGO PDUs using a Virtual Simulator Surrogate (VSS) and a Network Gateway (NG) was described in Pratt and Pratt [1996]. Figure 1 shows the VSS scenario in which one Blue Force (BLUEFOR) tank company follows a scripted path consisting of several segments and corresponding company formations. The shaded portions of the path indicate the relative geographic width spanned by the company units as they travel the path. For example, during the Road March the units are in a narrow single file formation, whereas during the Open Field Deploy segment, the units spread out and occupy a much wider area while travelling. In the actual implementation, we ended up using a 14 unit tank company instead of the original 18 unit tank company to be consistent with the BLUEFOR tank company generated by ModSAF.

Three static Opposing Force (OPFOR) units were placed at various distances from the path to force de-grouping. The hatched region shows the area in which entities will not be grouped together using the IGO PDUs because they lie within an arbitrarily specified radius of 4 km of OPFOR. Units in this region are de-grouped and described individually by PDUs until they move outside the hatched region. During these experiments units which become mobility impaired or catastrophically damaged are not re-grouped.

The VSS sends out Entity State (ES) or Entity State Update-1 (ESU-1) PDUs to describe the entities it controls. The NG groups together both Blue and Red force units as long as no relative enemy forces are within the degrouping radius. It performs simple linear dead reckoning of all the grouped entities and sends out a new IGO PDU if it receives an ES PDU or ESU-1 PDU for a grouped entity which exceeds some arbitrary dead reckoning tolerance in position or orientation (in our experiments, 1 meter and 3 degrees, respectively). IGO PDUs are also sent whenever the composition of a group has changed, e.g. entities are added or removed from the group. Otherwise, the NG sends out IGO PDUs at some regular heart beat interval, in our case once per five seconds. ES or ESU-1 PDUs from entities which are not grouped pass through the NG unmodified.

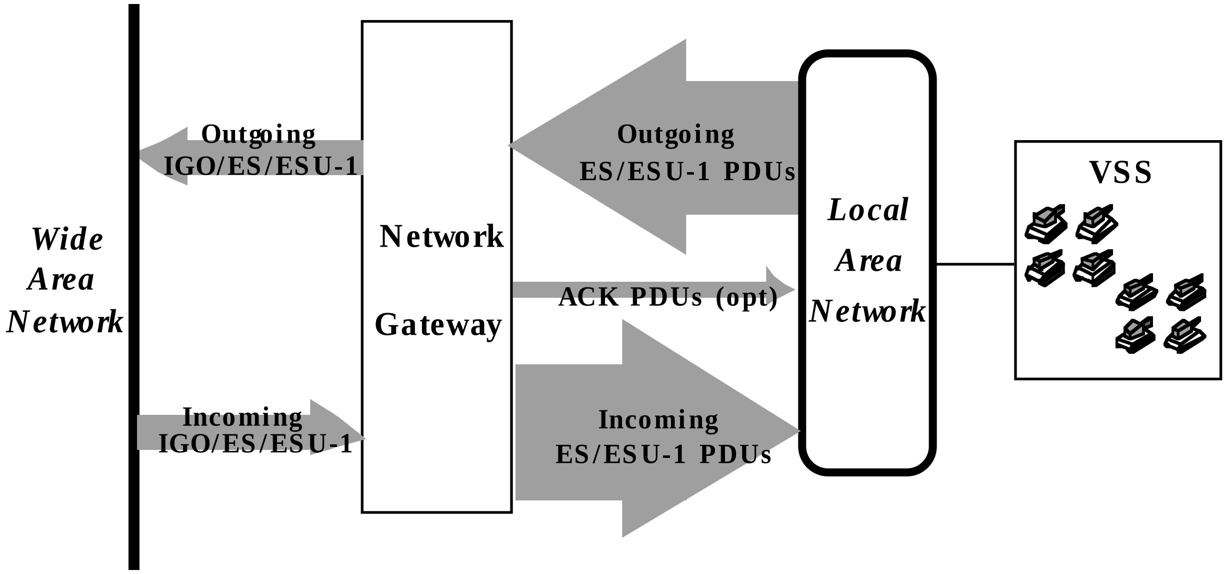

Figure 2. Network interactions resulting from using IGO PDUs

The prototype IGO PDU we used in our simulation experiments is defined in Table 1. It contains only minimal information to describe the group unit itself but has attached Grouped Entity Descriptions (GEDs) to describe each of the individual entities which comprise the group. The Group ID corresponds to the NG node site/address and which group is being represented. The Grouped Entity Category field is used to enumerate what type of GEDs are attached. The total number of entities comprising the group is given in the Number of Grouped Entities field. GEDs, which describe each of the entities in the group, contain nearly the same information as an ESU-1 PDU but in a more compact form. A prototype GED definition for describing ground vehicles in our experiments is given in Table 2.

NG nodes control all grouping of individual ESPDU information into IGO PDUs for transmission on WANs and degrouping of group information into individual ESPDUs for transmission within LANS. Thus, within each LAN the traditional DIS paradigm for exchanging ESPDU information remains unchanged. As a result, all intelligence for grouping resides with the NGs, and no modifications need to be made to existing simulations. Optionally, NGs can also send Acknowledgment PDUs to individual simulation nodes to advise of a grouped or non-grouped status, if needed. The resulting network interactions from implementing IGO PDUs are shown graphically in Figure 2. The thickness of the arrows indicates the relative network bandwidth used in sending out those PDUs in a scenario where entities were largely grouped.

RESULTS

Table 3 summarizes the various experiment runs, the number of the PDUs generated and the accompanying network bandwidth. To help establish a baseline for the amount of PDUs generated prior to implementing IGO PDUs, static runs (where all the entities did not move and only heart beat PDUs were being sent out regularly at five second intervals) and scenario runs were both made. Runs 1 and 2 used only ES PDUs while Runs 3 and 4 used both ES PDUs and ESU-1 PDUs. The PDU counts and bandwidth from Runs 1-4 represent the network traffic on the Local Area Network (LAN). Scenario runs with the use of IGO PDUs (Runs 5-8) were then conducted with OPFOR located at varied distances to show the bandwidth effects due to grouping and de-grouping. The PDU counts and bandwidth from Runs 5-8 represent the network traffic going out onto the WAN.

Our tests prove that the use of IGO PDUs can significantly reduce network bandwidth. When entities were fully grouped during the scenario simulation (Run 8), network bandwidth resulting from the use of ES PDUs only (Run 2) was reduced by nearly 90%. Similarly, network bandwidth resulting from the use of ES PDUs and ESU-1 PDUs (Run 4) was reduced by 80%. Runs 5, 6 and 7 had OPFOR located at approximately 3.25 km, 3.75 km and 4.25 km of the path. This lead to partial de-grouping of some of the BLUEFOR units during the simulation which resulted in higher individual entity ESU-1 PDU counts. Reduction in bandwidth was inversely proportional to the time BLUEFOR units were located within the de-grouping radius. Importantly, the grouped entities were able to be relatively accurately represented on a visual simulator which is sufficient for the needs of wide area sensors.

Table 1 and Table 2 describe the prototype IGO PDU we used for ground combat vehicles in our experiments. Only bearing and range was used to specify location because a fixed point of reference and ground clamping was assumed. In the general case, location needs to be flexibly specified in three dimensions and other vehicle specific information needs to be provided. Thus, we have proposed revised IGO PDU and IGO PDU GED definitions for ground combat vehicles, ground logistics vehicles, fixed wing aircraft, rotary wing aircraft, and individual combatants. Optional 32 bit extensions to each IGO GED have also been defined to track logistics information useful for both analysis and training purposes (Rieger, 1996). These Enhanced IGO PDUs contain information such as fuel/ammunition supply and maintenance statuses. Each IGO PDU will identify what type of Grouped Entity Descriptions are contained within it using a Grouped Entity Category field value [see the DIS Standards document for full details about IGO PDUs field definitions and field enumerations].

CONCLUSIONS

We have shown that use of the IGO PDU can significantly reduce the overall bandwidth of the PDU data stream onto a WAN. This facilitates higher echelon multi-site exercises allowing entity level representations while minimizing network overload between sites. This also encourages us to investigate use of lower bandwidth WAN network technology such as ISDN. ISDN is readily available commercially at most locations and is much less costly to use vice leasing time on Defense Simulation Internet (DSI).

As the community addresses the issues of filtering and data distribution, the IGO PDU and NG concepts deserve serious study. The work presented here deals with the development and implementation of DIS PDUs, but the same analysis and distribution mechanisms can be applied to other emerging network schemes and protocols. As the community transitions to an attribute based message passing scheme, the concept of a NG provides a means to maintain some backward compatibility with existing systems. Attributes passed as part of an IGO PDU can provide some guidance as to what information need to be exchanged.

In the next phase of this project, we hope to continue this work by exploring the possibility of keeping entities grouped together while engaged with opposing forces, that is, allow fire and detonation PDUs to be sent and processed while entities remain grouped. We also hope to experiment with some techniques to group entities in a hierarchical fashion via combinations of IsGroupOf and IsPartOf PDUs and perhaps allow mixed GED types to be contained within IGO PDUs.

ACKNOWLEDGMENTS

This research has being funded as part of the FY96 SIMTECH program.

REFERENCES

Foss, B. and Franceschini, R. (1995) “A Unified Aggregate Protocol.” *Proceedings of the 13th Workshop on Standards for the Interoper-*

ability of Distributed Simulations. Orlando, FL. September 1995.
Patterson, D. and Maxel, M. (1995) “The IsPartOf PDU Implementation Details.” *Proceedings of the 12th Workshop on Standards for the Interoperability of Distributed Simulations*. Orlando, FL. March 1995.
Pratt, S. and Pratt, D. (1996) “Use of the Is Part Of PDU for Unit Aggregation: Preliminary Report.” *Proceedings of the 14th Workshop on Standards for the Interoperability of Distributed Simulations*. Orlando, FL. March 1996.
Rieger, L. (1996) “IsPartOf PDU FY96 SIMTECH Program.” Presentation at 64th Military Operations Research Society Conference. June 1996.

Field Size (bits)	Field	Field Descriptions
96	PDU Header	IGO PDU Header
48	Group ID	Entity ID of NG group
8	Grouped Entity Category	Enumeration - 8 bit unsigned int
8	Number of Grouped Entities	8 bit unsigned int
Nx128	Grouped Entity Description	(See Table 2)

Table 1: Prototype IGO PDU for Ground Combat Vehicles (160 + Nx128 bits)

Field Size (bits)	Field	Field Descriptions
16	Entity	Entity - 16 bit unsigned int (Site & Application as in IGO Header)
16	Bearing from Reference Point	Milliradians - 16 bit unsigned int
16	Range from Reference Point	Meters - 16 bit unsigned int
16	Absolute Heading	Milliradians - 16 bit unsigned int
32	Appearance	Enumeration - 32 bit unsigned int
16	Speed	Meters/sec - 16 bit unsigned int
16	Turret Slew Rate	Milliradians/sec - 16 bit unsigned int

Table 2: Prototype GED (128 bits) for Ground Combat Vehicles

.

Run	Experiment	Grouping	ES PDUs	ESU-1 PDUs	IGO PDUs	Kbits/sec
1	Static	None	3745	0	0	5.77
2	Scenario	None	5115	0	0	7.88
3	Static	None	17	3741	0	3.35
4	Scenario	None	17	5111	0	4.57
5	Scenario	Partial	17	2696	218	2.85
6	Scenario	Partial	17	1918	324	2.32
7	Scenario	Near Full	17	30	408	0.93
8	Scenario	Full	17	0	405	0.90

Table 3: Results of the Simulation Experiments